AI Weapon Detection System for Real-Time Using Deep Learning with YOLOv8

1. Introduction

1.1 Purpose

The purpose of this project is to design and implement an AI-powered **real-time weapon detection system** using YOLOv8. The system detects firearms (pistols) and knives in live video streams and provides instant alerts to enhance public safety. This reduces reliance on manual surveillance, minimizes human error, and enables faster threat response.

1.2 Intended Audience and Users

- **Primary users**: Security staff in schools, malls, airports, banks, and public spaces.
- **Secondary users**: Researchers, law enforcement agencies, and developers working on AI-based security solutions.

1.3 Scope

The system leverages **YOLOv8** for high-speed, high-accuracy object detection. Trained on a dataset of **1,579 images with 1,735 weapon instances**, it classifies and localizes weapons in real time. The trained model achieved **mAP@50 = 0.875** and **mAP@50–95 = 0.596** after **50 epochs of training (~57 hours)**.
The system supports live CCTV/webcam input and will later be optimized for **edge deployment** (e.g., Jetson Nano, Raspberry Pi with accelerators).

1.4 Definitions, Acronyms, Abbreviations

- **YOLO**: You Only Look Once (object detection algorithm)
- **CNN**: Convolutional Neural Network
- **mAP**: Mean Average Precision
- **FPS**: Frames Per Second
- **TPR**: True Positive Rate
- **NMS**: Non-Max Suppression

2. Overall Description

2.1 Product Perspective

The system is a **standalone AI surveillance module** that integrates YOLOv8 for detection and a custom alert system for notifications. It processes real-time video, detects weapons, and triggers alerts through on-screen warnings and optional SMS/Email APIs (e.g., Twilio).

2.2 Product Functions

- Capture live input from CCTV/webcam.
- Detect weapons (knife, pistol) in real time.
- Apply a confidence threshold ≥ 0.7 before confirming detections.
- Display bounding boxes and labels with confidence scores.
- Send real-time alerts (screen notification, optional SMS/email).
- Maintain detection logs for analysis.

2.3 User Characteristics

- Users may not have technical knowledge.
- The interface must be **simple, responsive, and reliable**.
- Minimal user interaction required (just monitoring).

2.4 Constraints

- High FPS requires GPU acceleration (CPU-only performance is limited).
- Training dataset limited to **knives and pistols** (future classes may be added).
- Requires stable power and camera input for continuous monitoring.

2.5 Assumptions and Dependencies

- Deployment environments will use either **workstations with GPUs** or **edge devices** with accelerators.
- Python libraries (TensorFlow, PyTorch, OpenCV) will be available.
- Alert notifications depend on third-party APIs (e.g., Twilio).

3. Specific Requirements

3.1 Functional Requirements

- **FR1**: Capture video from CCTV/webcam in real time.
- **FR2**: Detect knives and pistols with YOLOv8.
- **FR3**: Show bounding boxes, labels, and confidence score ≥ 0.7.
- **FR4**: Trigger alerts (on-screen + optional SMS/email).
- **FR5**: Maintain detection history logs.
- **FR6**: Operate offline once deployed.

3.2 Non-Functional Requirements

- **NFR1**: Performance → ≥10 FPS on GPU; ~17 FPS achieved on CPU in testing.

- **NFR2**: Accuracy → Achieved mAP@50 = 0.875, Recall = 0.807, Precision = 0.871.
- **NFR3**: Usability → Must have a simple, minimal interface.
- **NFR4**: Reliability → Continuous monitoring with stable detection.
- **NFR5**: Scalability → Extendable to multiple CCTV feeds.
- **NFR6**: Security → Protect detection logs and alerts from tampering.

3.3 External Interface Requirements

- **Camera**: USB/webcam or CCTV feed.
- **Alert System**: Desktop notification, optional SMS/email API.
- **Software**: Python (YOLOv8, OpenCV, TensorFlow/PyTorch, Scikit-learn).

4. System Features

4.1 Real-Time Weapon Detection

- Detects **knives and pistols** from video feed.
- Trained on a custom dataset (1,579 images, 1,735 instances).
- Uses YOLOv8 with **50 training epochs** and fine-tuned hyperparameters.

4.2 Alert & Monitoring System

- Generates real-time alerts with bounding boxes.
- Displays object class and confidence score.
- Optional **SMS/email notifications**.

4.3 Offline Operation

- Can function without internet (for detection).
- Alert APIs may require internet (optional feature).

5. Other Requirements

5.1 System Limitations

- Current model limited to **knife and pistol detection**.
- Recall for pistols is slightly lower (0.735) compared to knives (0.884).
- Edge deployment on devices like **Raspberry Pi 4** will have lower FPS unless optimized (e.g., with TensorRT or TFLite).

5.2 Future Scope

- Expand dataset to include more weapon types (rifles, shotguns).
- Multimodal fusion with **thermal or radar data** for concealed weapon detection.
- Integration with **smart surveillance dashboards**.
- Edge optimization with **quantization and pruning** for Jetson/RPi deployment.
- Cloud-based central monitoring across multiple locations.